GENI Federation
Software Architecture Document

Version 1.0 March 2012

GENI Architecture Team
Marshall Brinn and Rob Ricci, co-chairs

Introduction

This document presents the software architecture of the GENI (Global Environment for Network Innovations) Federation. This description, though simple, calls for some introductory points of clarification and scope.

The term *GENI* has come to mean many things in different contexts:

- GENI is, first, a *vision* for providing and managing highly scalable programmable dynamic computation topologies from a broad range of heterogeneous resources.
- GENI is also the name of the NSF *program and community* of researchers, campuses, architects, and developers working to bring the GENI vision to fruition.
- GENI is a *software architecture* defining interfaces by which such topologies are managed and used and such resources are federated.
- GENI is the name of a specific *federation ("the NSF GENI Federation")*, built under the GENI program, which represents an *implementation and deployment* of the GENI architecture for the GENI program on a specific set of hardware resources and a set of *policies* for administering those resources in the context of that federation.

By these terms, this document describes the *software architecture* of *the GENI Federation*. As such, it describes both those aspects of the software architecture that apply to "a *GENI federation"* (i.e. any federation that is built on the GENI vision and software architecture), as well as those aspects that are specific to requirements of the "*NSF GENI Federation"*. It does not describe the hardware architecture (i.e. the specifications of computation and networking hardware) nor does it describe all of the policies, operations and agreements required in managing federated resources.

We will focus on addressing several key questions:

- *Services*: What are the software services that constitute the GENI Federation?
- *Interfaces*: What are the interfaces presented by these services for use by other GENI services or by external clients?
- *Interactions*: What are the underlying interactions and relationships among entities that characterize key use cases? Critically, which relationships among entities reflect *trust* (e.g. an exchange of key materials)?
- *Requirements*: What specific requirements does the GENI Federation impose on the software architecture and how are these requirements addressed?

This document is the product of the GENI Architecture Team[1], a group of leading architects from across the GENI community, which is tasked with defining the software architecture of the GENI Federation.

[1] The team membership as of GEC13: Marshall Brinn (GPO) and Rob Ricci (University of Utah), co-chairs, Nick Bastin (Stanford University), Jeff Chase (Duke University), Max Ott (NICTA), Larry Peterson (Princeton University) and Chip Elliott (GPO), voting members.

GENI Vision

The GENI vision can be rendered in many ways, but this brief statement may suffice for the purpose of introducing a description of the architecture:

> GENI is a *deeply programmable infrastructure suite* for performing computer science experimentation *at scale*.

Drilling down slightly may help clarify some of the terms of this description:

- *Deeply Programmable*: Allows programmatic configuration and control of all aspect of a computation network (computation, storage, communications).
- *Infrastructure suite*: Provides transparent access to a federation of sliceable and shareable resources in programmable topologies.
- *At Scale*: Allows support for extremely large, distributed experimental topologies, distributed across 100-200 campuses with realistic traffic loads or real traffic from users who 'opt in'.

Architectural Overview

This section introduces the top-level software architectural requirements and entities. More details for each of these will be provided in subsequent sections.

We begin by defining the concept of federation itself.

> A *federation* is a set of agreements among people or organizations, representing the policies and terms under which they will trust, collaborate, share resources or engage in other common activities. A *federation architecture* is a set of *constructs and services* that codify and enforce those agreements.

The GENI software architecture seeks to mediate the requirements of a set of different (human) principals:[2]

- *Resource Owners* that own computation and network resources at a campus, test-bed, regional or backbone network that they are willing to contribute to the GENI federation for use subject to policies they set. In addition, we consider requirements of *Operations Staff* who manage the operations of these resources.
- *Experimenters* who wish to make use of federation resources for conducting experiments, and deploying services or applications. Note: this group may also

[2] Another set of GENI participants whose requirements the architecture plans to meet is the 'Opt-In User', which will be discussed in future revisions of this document.

include researchers, students, application developers or other resource consumers, but the main focus of the GENI federation is experimenters.

The GENI software architecture represents the requirements and capabilities of these sets of principals in software:

- *Aggregates* are software representations of the resources available for sharing within the Federation. Specifically, an Aggregate is any software entity that implements the GENI Aggregate Manager API (AM API) to make its resources (computation, network, storage, etc.) accessible to the Federation.
- *Experimentation Tools* support the requirements of Experimenters in composing topologies from federation Aggregates for deploying and conducting experiments.

Building a federation among aggregates and researchers requires two additional software constructs:

- *Identity Provider*: A mutually trusted entity to sign certificates that authenticate a given person to the federation as having particular attributes. A GENI federation requires its members to trust at least the same set (one or more) of Identity Providers.
- *Credential Provider*: A mutually trusted entity that provides particular credentials to authorize particular actions (e.g. allocation of resources by the AM API). The provider "vouches for" the experimenter and bears some accountability for actions taken by the experimenter. GENI calls one such provider a *Slice Authority* and the credentials it provides *Slice Credentials*.

Figure 1 presents the top-level interactions among these software entities. It reflects, in fact, the state of the implemented GENI federation as of this document revision (circa GEC13, 3/2012).

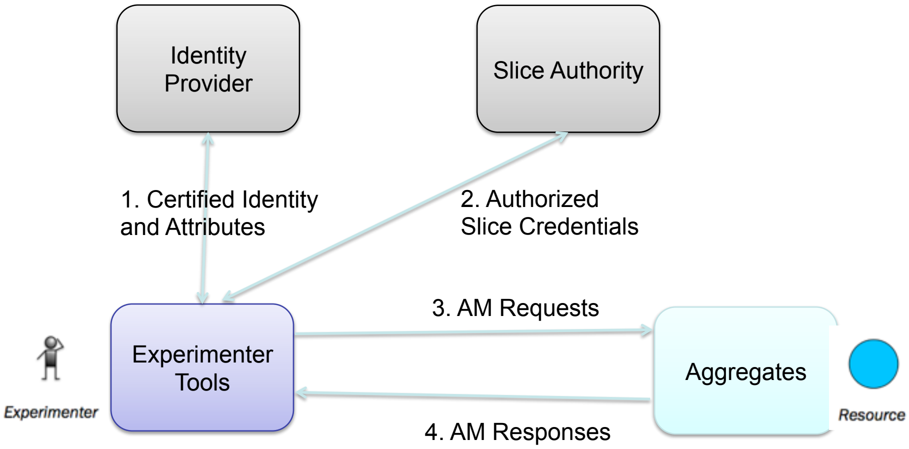

Figure 1. Generic GENI configuration showing SA, IdP and direct interaction between Experimenter Tools and Aggregates.

Any federation of resources that conforms to the GENI software architecture can be considered as a GENI federation. However, the NSF GENI Federation, the specific instance of this architecture built as part of the NSF GENI project, entails additional requirements and principals.

Different principals enter into a federation with different goals and requirements. The Experimenter wants *access to resources* from which topologies can be built on which to run experiments. The Operations Staff and Resource Owners are concerned with *protecting their resources* against inappropriate use, which could damage the resources or make them liable in some way.

The NSF GENI Federation seeks to provide the functionality desired by Experimenters while providing the protection assurances desired by Resource Owners and Operations staff. Specifically,

- The NSF GENI Federation will provide specific *authentication services*, so that transactions between software entities (representing physical entities) are made on the basis of mutual trust of a third party. In this way, Researchers and Aggregates can become active, trusted members of the federation.
- The NSF GENI Federation will provide specific *authorization services*, so that a principal may not take an action that is prohibited by Federation policies.
- The NSF GENI Federation will provide *accountability services*, so that there is a principal (e.g. a project lead) who is responsible for all actions taken on resources allocated on that principal's behalf.

Reflecting these requirements, the NSF GENI Federation adds an additional set of principals and software representatives, namely:

- *GMOC* (GENI Meta-Operations Center) Staff oversee the health and proper operations of the NSF GENI Federation. GMOC staff are supported by GMOC tools to meet the requirements of GMOC operations.
- *Clearinghouse* provides the required trust, authentication, accountability and authorization services for the NSF GENI Federation*.*

Figure 2 presents the top-level dataflow interactions among the top-level entities comprising the NSF GENI Federation. The diagram intentionally does not indicate whether Logging and Authorization takes place internal or external to the Aggregates (by some federation-level or mutually trusted third party service). As we shall describe, the architecture supports different configurations deploying these functions in different ways, subject to mutual agreement by the Federation and the Aggregate.

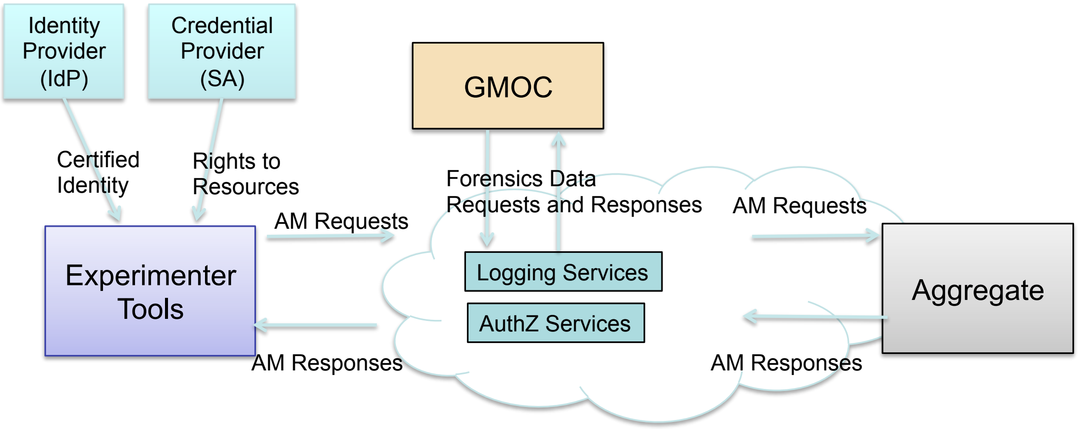

Figure 2. Dataflows among the three principal clients of GENI services within the NSF GENI Federation: the GMOC, the Experimenter Tools and the Aggregates. Every AM API transaction between the Experimenter Tool and Aggregate may be subject to Authorization and Logging. The "cloud" indicates that the locus of the Logging and Authorization may be internal or external to the Aggregate.

Software Architecture Details

The GENI architecture reflects a hybrid approach towards federation. GENI reflects a belief in the autonomy of aggregates, and local decision and control regarding resource management and provision of services to clients. It is also explicitly designed to allow aggregates to participate in more than one federation simultaneously. The GENI Federation seeks to manage issues of identity and authorization at a federation level to provide a degree of protection and assurance to all members of the federation. The GENI architecture also reflects a similar hybrid philosophy towards monitoring and maintenance, by which federation services are informed by aggregates on the state they choose to share, from which high-level views and decisions may be taken. Finally, GENI seeks to be expansive to the inclusion into the federation of new resource types at new or existing aggregates.

The remainder of this section provides details on the software entities comprising the GENI Federation Architecture and their interactions.

Trust Relationships

The GENI Federation rests on trusted interactions among the different entities (people and the tools or services that represent them). There are essentially two kinds of trust statements required for these interactions: *authentication* (identifying the requester) and *authorization* (certifying the attributes of the requester).

Any decision to honor a connection or request from another entity requires both of these kinds of trust: I need to believe that this is who they claim to be, and can then decide whether they have the rights to take the action they are requesting based on applicable policy.

Trust within GENI is based on the exchange of PKI key materials, and a trusted statement is one signed by a trusted entity with their private key. If I have access to the public key of a trusted entity and can validate the signature of a message signed by that entity with their private key, I can be assured that they signed (and thus believe or assert) that statement.

Every GENI federated AM must import the set of GENI root certificates and accept credentials signed by any GENI Identity Provider (IdP). Additionally, a GENI federated AM may optionally trust root certificates of other GENI federation members, and accept credentials signed by these members. From these trust foundations, we may establish chains of trust: if I trust any statement signed by a root, I may trust any statement signed by an entity with particular attributes asserted by that root, etc. For example, if the federation trusts a university registrar, all statements of membership in a given class made by that registrar may be trusted within the federation.

Beyond trust of the GENI root, there are additional kinds of trust established between entities by which one entity (A) authorizes or permits another (B) to act:

- *Delegation*: By delegating a privilege or right to entity B, entity A trusts B to perform tasks that A is currently permitted to do. B is then free to act autonomously with this new privilege based on that trust. This delegation is performed by means of an additional credential, signed by A, that B has a particular attribute or right. Note that this is a general delegation, not typically given on a case-by-case basis.
- *Representative*: For more granular control, A can assert that B 'speaks for' A in a particular request. A must delegate to B the general privilege that he 'may speak for' A. A specific request, signed by B, can be said to be made on A's behalf. Typically this request will be tagged with additional attributes (a 'speaks-for' tag e.g.).

Aggregate Services

Before we describe the aggregate services, some definitions are in order.

An *aggregate* represents a resource or set of resources that can be offered for inclusion in the topology specified for an experiment. These resources may fall into broad categories including:

- *Computation resources*: Resources that provide programmable operating environments, such as a discrete running operating system (virtual or genuine) or a shared runtime-environment (e.g. thread based). This category also includes sensors, or mobile handsets.
- *Communication resources:* Resources that provide access to resources that control, divert, modify or shape traffic at different layers (e.g. an OpenFlow switch, Ethernet, software routers or network processors).

- *Storage resources:* Resources that provide access to permanent or temporary data storage for a given experiment or application.
- *Sensing and Actuating resources:* Resources that provide real-time measurements from or take action in a physical environment (e.g. radars, robots, radios).

Aggregates provide *slivers* of their resources to requesters through invocations of the AM API. These slivers may be a whole physical resource (e.g. a bare metal machine), or a virtualized piece of a resource (e.g. a virtual machine), or a combination of these.

Different aggregates may provide different levels of performance isolation for different slivers on the same resource.

A client of aggregate services (e.g. an experimenter or tool working on the experimenter's behalf) gathers slivers provided by aggregates in a grouping called a *slice*. The slice is primarily an accounting mechanism, allowing for determining which resources are allocated to which clients for a particular purpose.

Slices are associated with a *project*, allowing another layer of accounting: there are multiple projects, which may each have multiple slices, which may have multiple slivers. A project has a single person, the Lead, associated with the project who is root of accountability for all activity on all slivers associated with slices of that project. An experimenter may have multiple slices or be a member of multiple projects, but individual actions taken by the experimenter are done in the context of a single slice (which is in a single project) for accountability.

The slice may contain slivers from a single aggregate or may span across multiple aggregates. In the latter case, an operation called *stitching* may be required to ensure that shared constraints such as common services or network connections (e.g. VLAN identifiers) are shared across different aggregates and slivers. Put another way, stitching is a mechanism for joining various aggregate-local topologies into a single cross-aggregate topology.

The AM API is defined in greater detail at http://groups.geni.net/geni/wiki/GAPI_AM_API. The essential service types that an aggregate provides are summarized here:

- *Directory services*: Listing resources available at an aggregate.
- *Reservation services*: Pre-allocating resources, which may be useful in supporting cross-aggregate allocations.
- *Allocation services*: Requesting a sliver or list of slivers from an aggregate manager to be placed into a slice.
- *Sliver services*: Requesting changes to a sliver's state (modifying its current life-cycle stage, e.g.)

The AM API is supported by any Aggregate in a GENI Federation. The details of what resources it may provide and what request options it recognizes vary from aggregate to aggregate and over time. Similarly, we note that the AM API provides a programmatic interface but this interface contains a significant declarative component in the form or RSpecs (Resource Specifications) that are passed as part of the API call (*request* RSpecs) or response (*advertisement* or *manifest* RSpecs). These also have elements that are universal and others that are resource or even aggregate specific.

Federation Services

A GENI federation provides a set of interrelated services to support the requirements of Experimenters and Aggregates as they conduct AM API transactions. The services will be described separately for generic GENI federations and for the NSF GENI Federation in the following subsections.

Federation Services: Generic GENI Federation

A Generic GENI Federation is composed of a few simple building blocks:

- The Federation provides at least one Identity Provider that is trusted by all Experimenters and Aggregates. All interactions may thus be *authenticated* through aggregate-specific mechanisms.
- The Federation may provide Credential Providers that generates credentials that are signed by an authority that is trusted by all Experimenters and Aggregates. All interactions may thus be *authorized* through aggregate-specific mechanisms.
- No federation-level accountability services are specified for the Generic GENI Federation. Each aggregate logs whatever data it chooses for its own purposes.

The GENI Software Federation Architecture also provides a series of generic building blocks and patterns called *Coordinators*, which seek to support the identified and anticipated requirements of specific federations. These coordinators facilitate some degree of cross-aggregate control and visibility into the federation. They may (at least):

- Gather aggregate-level information into federation-level views
- Distribute federation-level (cross-aggregate) information
- Interpose between AM API calls between experimenters and aggregates.

Federation Services: NSF GENI Federation

The NSF GENI Federation provides a particular set of Authentication, Authorization and Accountability Services to support the particular requirements of this federation. In the context of the NSF GENI Federation, we refer to this set of services as the *Clearinghouse*.

Authentication Services. The Clearinghouse represents a trust anchor for all software entities (tools, aggregates, services) in the NSF GENI Federation. Any entity trusted by the NSF GENI Federation is trusted by any member of the NSF GENI Federation. Being trusted by the NSF GENI Federation means having credentials signed by GENI's PKI private keys testifying to particular attributes for that entity. The installation of the GENI certificate as a trust root at any GENI service allows for federated trust across people, aggregates and services. In this way, we do not need each entity to explicitly trust each other entity to allow for federation-wide trust: we only need each entity to trust GENI. Thus, we reduce the number of trust relationships from $O(N^2)$ to $O(N)$.

> Establishing trust is a human activity, involving agreements and understanding of one-another's procedures and requirements. Trust between software entities should be a reflection of trust shared by corresponding physical entities.

The Clearinghouse provides a series of services for managing and asserting the credentials of entities trusted by GENI.

- *Identity Providers* (IdPs) authenticating people or services to the GENI federation. The NSF GENI Federation provides its own IdP, while also trusting authentication from trusted IdPs outside the Federation such as from InCommon[3], or the IdPs of certain aggregate control frameworks.
- A *Service Registry* provides experimenters with a 'yellow pages' of URL's of all trusted services of different kinds. In particular, the list of all available aggregate managers trusted by GENI (possibly satisfying particular search criteria) is provided.
- A *Portal* provides web-based authentication and access to the authorized Clearinghouse services and other GENI tools, providing a per-session repository for user certificates and credentials for the user's convenience.

Authorization Services. The Clearinghouse provides services to support policy-decision points and policy-enforcement points with respect to Federation policies within the Clearinghouse or within Aggregates. All aggregates within the NSF GENI Federation must enforce federation policies, and may enforce aggregate-local policies as well.

There are two essential types of authorization policy we consider: *Trust Policy* and *Resource Allocation Policy*.

Trust Policy is a statement or sequence of statements from which allowable actions may be inferred from the attributes of a principal. These statements are credentials (assertions about a principal signed by a trusted authority) regarding possession or

[3] http://www.incommon.org.

delegation of rights and privileges, or e.g. a principal's role with respect to a particular project, slice, resource or institution.

Resource Allocation Policy is a statement limiting the resource allocations or allocation behaviors associated with a given project, slice or experimenter. For example, we may wish to limit the number of compute nodes (computers or VM's) allocated to a given project at any given time.

The NSF GENI Federation Clearinghouse provides an Authorization Service that determines whether a given action is permitted by policy. The Authorization Service contains a series of guards, each of which may veto a given action (i.e. an act is authorized if and only if it is allowed by every guard). Version 1 of the NSF GENI Federation Clearinghouse will include a trust guard and a resource allocation guard. As noted, an Aggregate may use this Authorization Service, or by mutual agreement, some other trusted authorization service.

The Clearinghouse provides a Credential Store that provides access to all credentials for all GENI-trusted entities. This store allows for federation or local authorization services or other policy decision or enforcement points to have access to the appropriate credentials without needing to carry or compute these at the time of each customer request. Aggregates may use this store, or provide their own internal credential maintenance. The Credential Store allows for mapping a known certificate or other unique identifier to a list of signed credentials associated with that individual. By keeping authorization credentials separate from authentication certificates and by imposing short expiration times on credentials, it is possible to modify credentials and have the effects of these modifications take effect in a reliable and timely manner throughout the federation.

The Clearinghouse provides two different Credential Authorities to generate and manage credentials of principals to act in different contexts:

- A *Project Authority* asserts the existence of projects and the roles of some members (e.g. Lead, Experimenter).
- *Slice Authorities (SAs)* providing experimenters with slice credentials by which to authorize AM API calls on federation aggregates. There may be other SAs that are trusted but not provided by the federation.

Accountability Services. The NSF GENI Federation requires that all AM transactions be logged for GMOC forensics. To support this requirement, the Clearinghouse provides services that log transactions (successful or failed) between experimenter tools and aggregates to support real-time and post-facto forensics analysis. This logging service fronts a store for writing and querying data associated with transactions, allowing for determining what entity made what requests and got what results. As noted, an Aggregate may use this logging service, or by mutual agreement, some other trusted logging service.

By having access to logs and databases of transaction callers and arguments, of projects and their slices and slivers, the GMOC can have critical timely trace back to find the identities of possibly misbehaving experiments or responsible project leads. They can then, depending on the situation, contact the project lead, shut down all or some slivers associated with a misbehaving aggregate or experiment, or some combination thereof. Additionally, these logs will be maintained for long periods to allow for long-term forensic analysis.

The logs at the GMOC (collected from the Federation and Aggregate logging services) provide the traceability between slivers and slices. The Slice Authority provides the link of slices to projects, while the Project Authority provides the link of projects to project leads. Together, these provide the ability to find the responsible party to contact in case of problematic behavior on the part of an experimenter.

Figure 3 provides a view of the services comprising the GENI Clearinghouse. The clients of these services, the Experimenter tools, aggregates and GMOC, access them via Clearinghouse API's and the GENI web portal, authorization permitting.

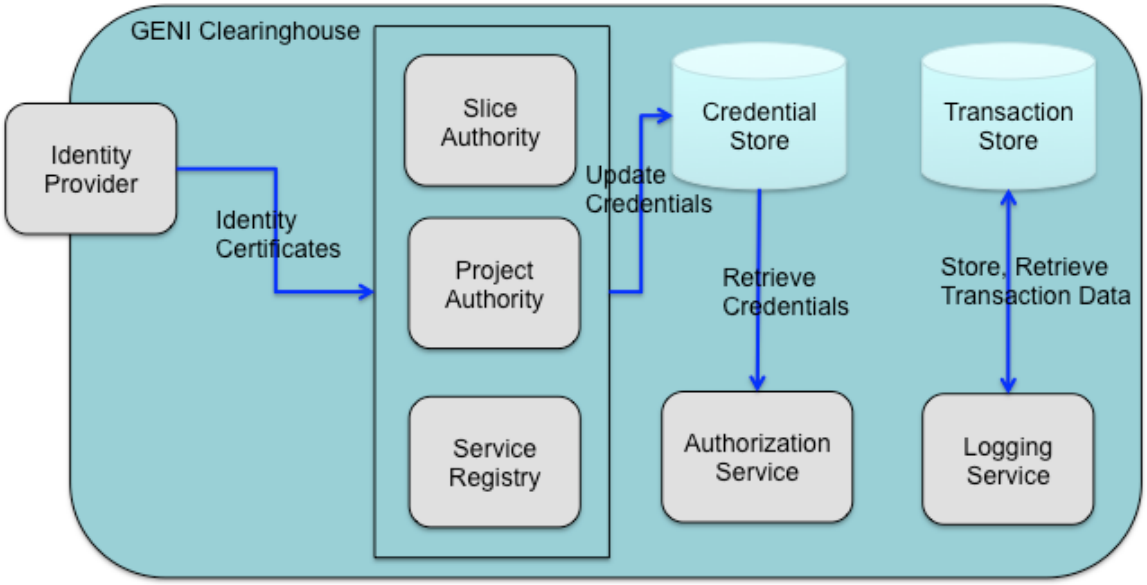

Figure 3. Essential GENI Clearinghouse services and their internal interactions. Note the IdP is depicted as slightly outside the CH since its operations are largely human-in-the-loop.

NSF GENI Federation Deployment Configurations

As discussed above, the accountability requirements of the NSF GENI Federation necessitate that all AM transactions between Experimenter Tools and Aggregates must be logged and authorized. An aggregate must use logging and authorization services the federation trusts, but these services may be private (internal to the aggregate) or public (external to the aggregate). Authorization must use federation policies, and may include aggregate-local policies as well. Public services may be supplied by the federation (as a "Clearinghouse" service) or by a trusted third party.

These approaches represent *deployment decisions*: the act of trusting an aggregate entails trusting the details and consequences of its deployment decisions.

Part of the federation agreement between an aggregate and the federation is that the federation accepts the aggregate's deployment decisions, including its choice of logging and authorization services. In that spirit, the NSF GENI Federation will consist of a range of different aggregates with configurations reflecting different approaches towards authorization and logging as well as resource management.

The GENI Software Architecture seeks to be flexible to support inclusion of a broad range of resource provider configurations and behaviors. We provide a range of deployment options, which are transparent to both experimenters and aggregates, for configuring an aggregate into the NSF GENI federation:

- *Local Trusted Services*. One configuration is that the Aggregate Manager (AM) has its own aggregate management infrastructure and provides its own authorization and logging services. By virtue of the trust implicit in federating with these aggregates, GENI trusts these local (aggregate-specific) services as well.
- *Federation Trusted Services*. Another deployment configuration is for aggregates to use the Clearinghouse Logging and Authorization services in the course of responding to any AM API request. This approach provides low overhead and low barrier to entry for contributed aggregates.
- *Local Proxy Aggregate Manager*. In this configuration, one AM acts as a pass-through wrapper for another AM, providing its own authorization and logging services. Such a configuration may be appropriate for configurations that don't provide their own authorization or logging services.
- *GENI Proxy Aggregate Manager*. This configuration is much like the local proxy, with the difference that there is a single proxy AM provided by the NSF GENI Federation. It passes through AM API requests, but logs the transaction and applies federation authorization policy 'in-the-loop'. Client tools are not permitted to talk to the individual AM's directly. The primary benefit of such an approach is that it allows Federation-wide policy to be applied in a central location, enabling resource allocation policies that span multiple AM's. It also allows closer control on revoked privileges, and catching and preventing disallowed actions before they happen.

Figure 4 illustrates the "Federation Trusted Services" configuration described above. Illustrating the other cases would require simple variants on this figure.

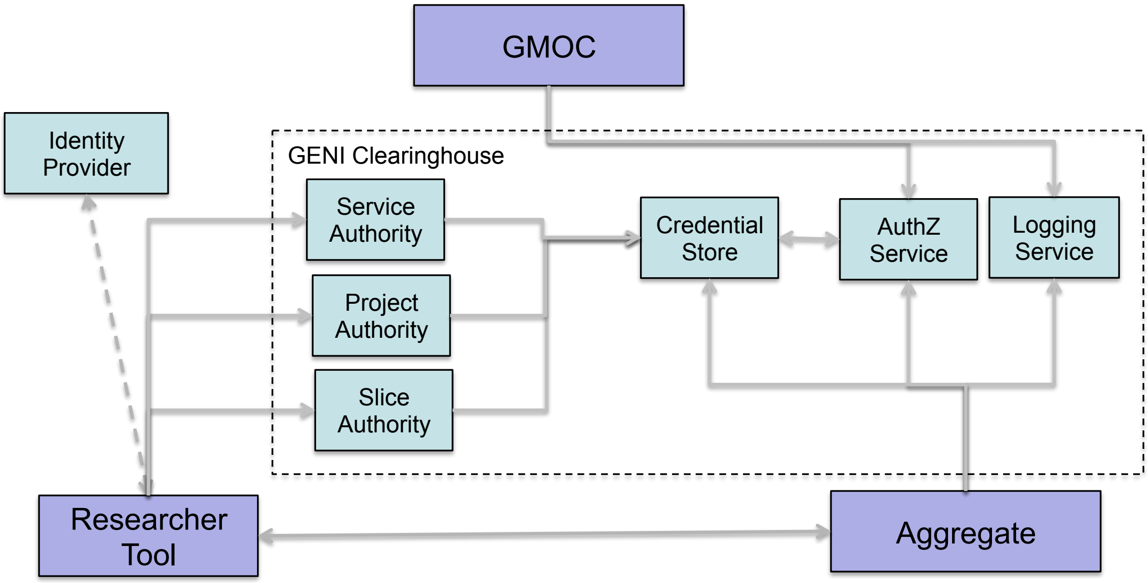

Figure 4. A Federation Trusted Services configuration representing a GENI Aggregate using Authorization and Logging services provided by the Clearinghouse.

Certain federation-level functionalities represent information that spans across aggregates and are thus managed by *coordinators.* An example of a coordinator would be a slice tracker or project tracker, which maintains the number of allocations of compute nodes to particular slices or projects to support federation resource allocation policies. Another example would be supporting particular alerts or analytics required by the GMOC.

NSF GENI Federation Trust Relationships

Overall, the NSF GENI Federation trusts people who have established their identity through a trusted IdP. By policy, the NSF GENI Federation provides particular trust and authority to faculty and leads of known projects to manage the delegation of authority within their projects. GENI trusts AM's who have established their attributes and identity through the federation Service Registry. The NSF GENI Federation trusts the services of the Clearinghouse. Any entity that initiates a GENI API call must have valid certificates and credentials for the recipient of the API call to authorize and authenticate the identity of the caller and the validity of the call.

Figure 5 illustrates the trust relationships underlying the NSF GENI Federation configuration.

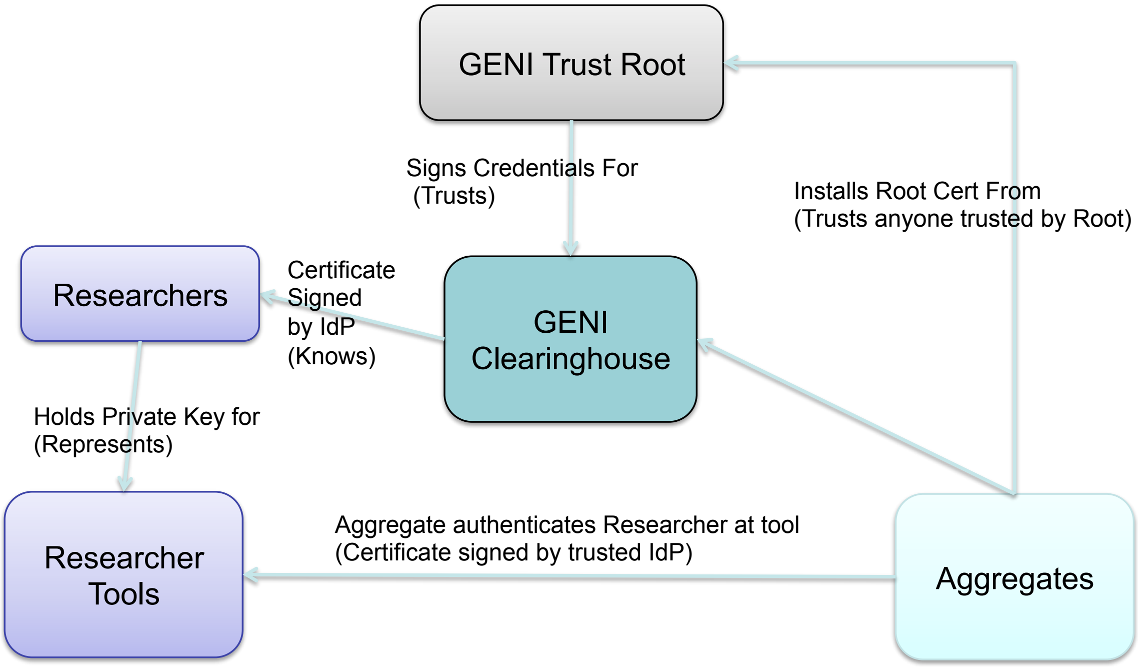

Figure 5. Trust relationships among key elements of the NSF GENI Federation configuration. The direction of the arrow indicates the trusted entity.